



09867183-052901

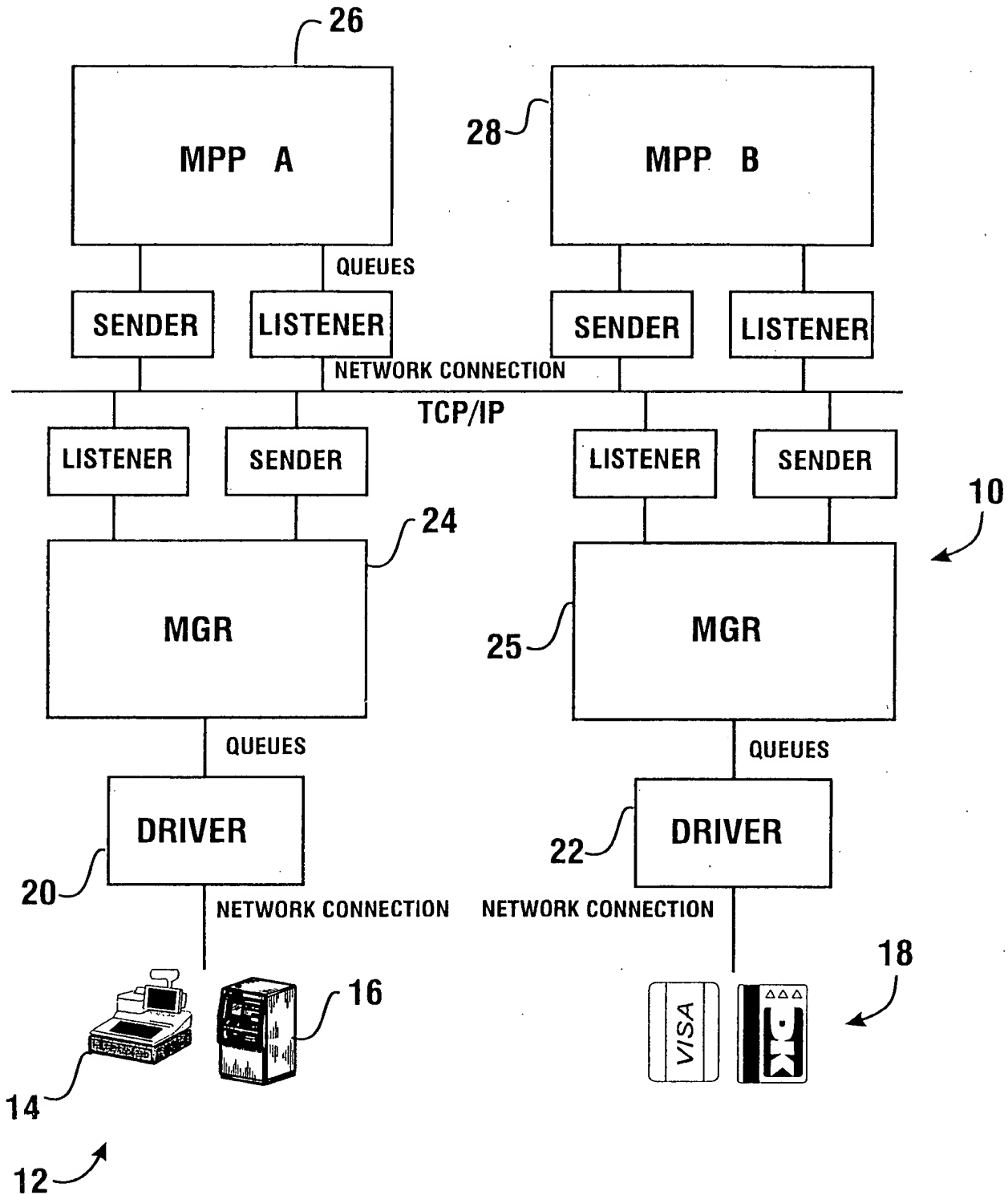
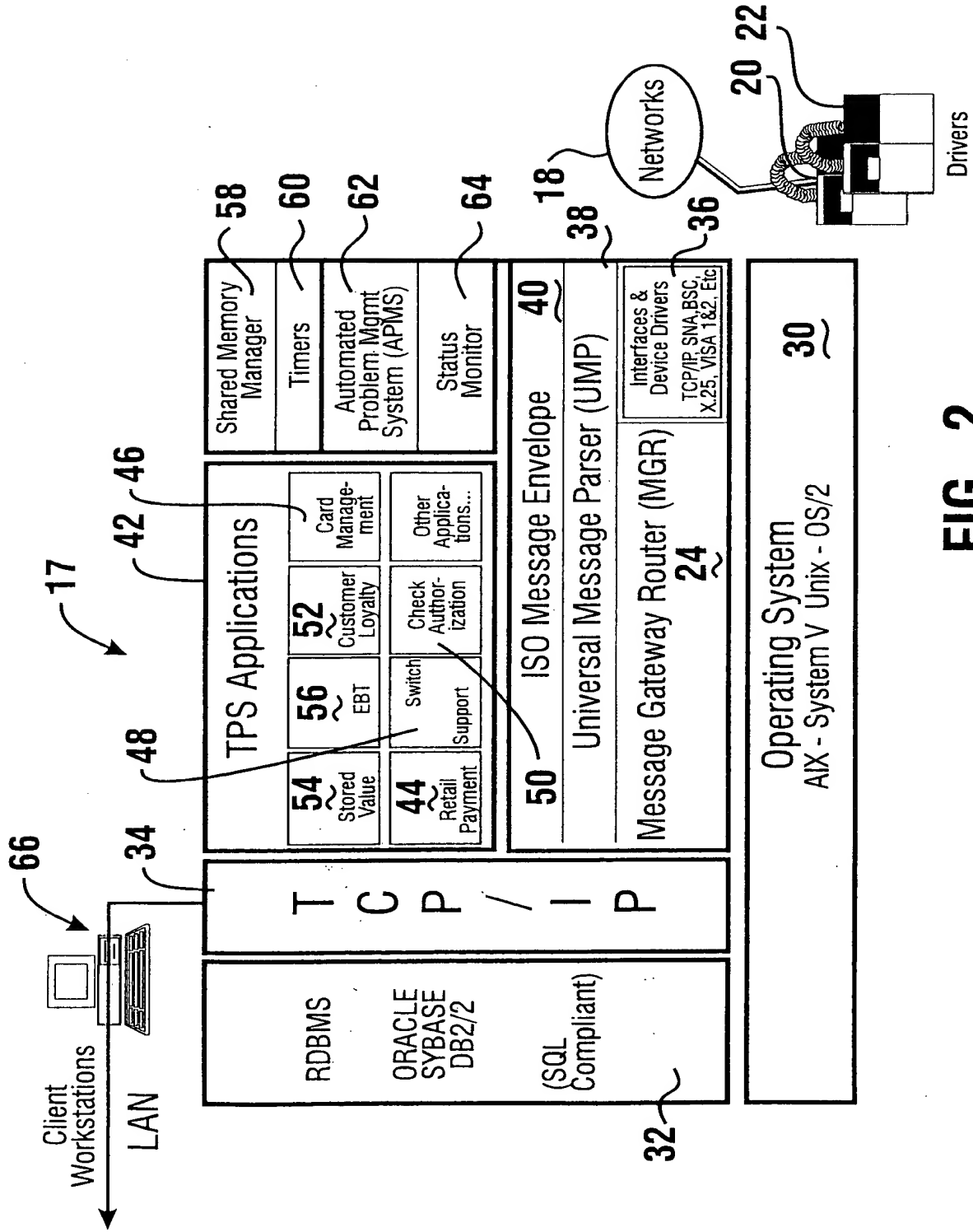


FIG. 1

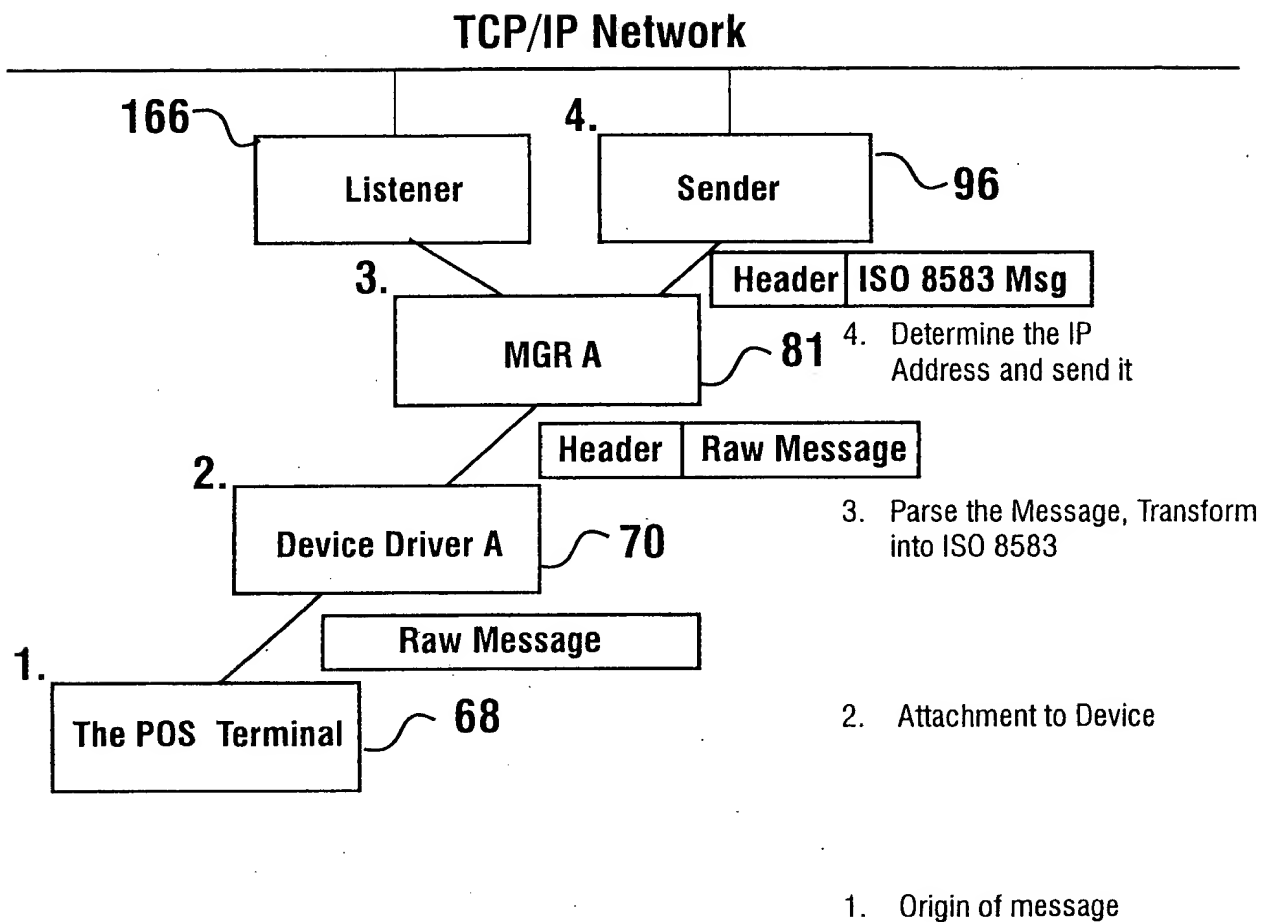


**FIG. 2**

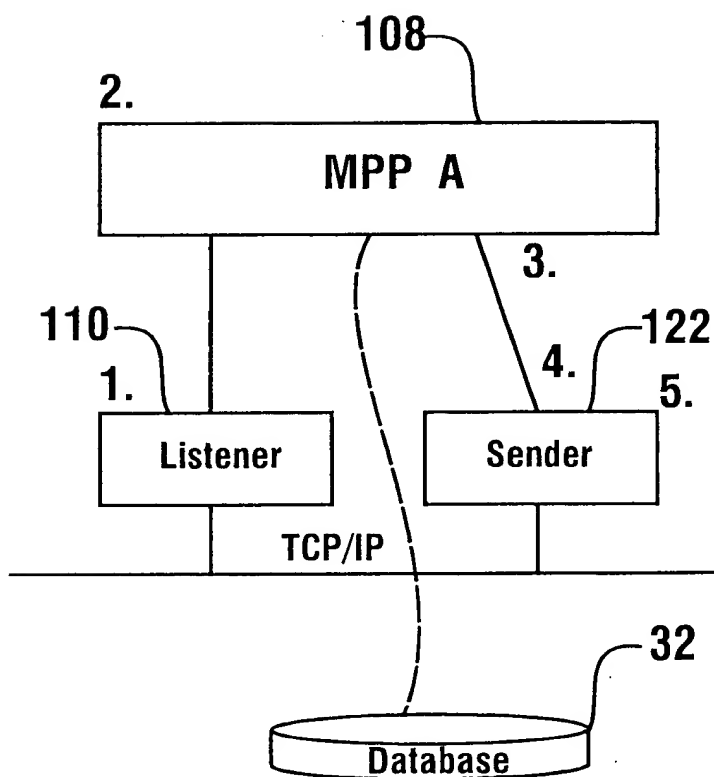
## Standard Message Envelope (SME) Format.

1	Header Sid	Header Layout Version	1
2	Source Node Sid	The message originating node system Id.	6
3	Message Receive System Time	The System time in YYYYMMDDHHMISSmmm format.	17
4	Internal Message Sid	Unique system Id of the received message.	4
5	Service Sid	The Message Processing Program (MPP) service system Id, which can process received message.	4
6	Target Node Sid	The message receiving node system Id	6
7	Data Format Indicator (Source)	Message data format type 0 - External Data Source 1 - Internal Data Source	1
8	Message Direction	The direction of message routing.	1
9	Processing Time	Elapsed message processing time in milliseconds.	5
10	Processing Node Sid	The last processing node system Id	6
11	Target Line Node Sid	Line driver node system id. Assigned when terminal is attached to line group.	6
12	Message Text	The Message text in ISO8583 format	Variable

**FIG. 3**

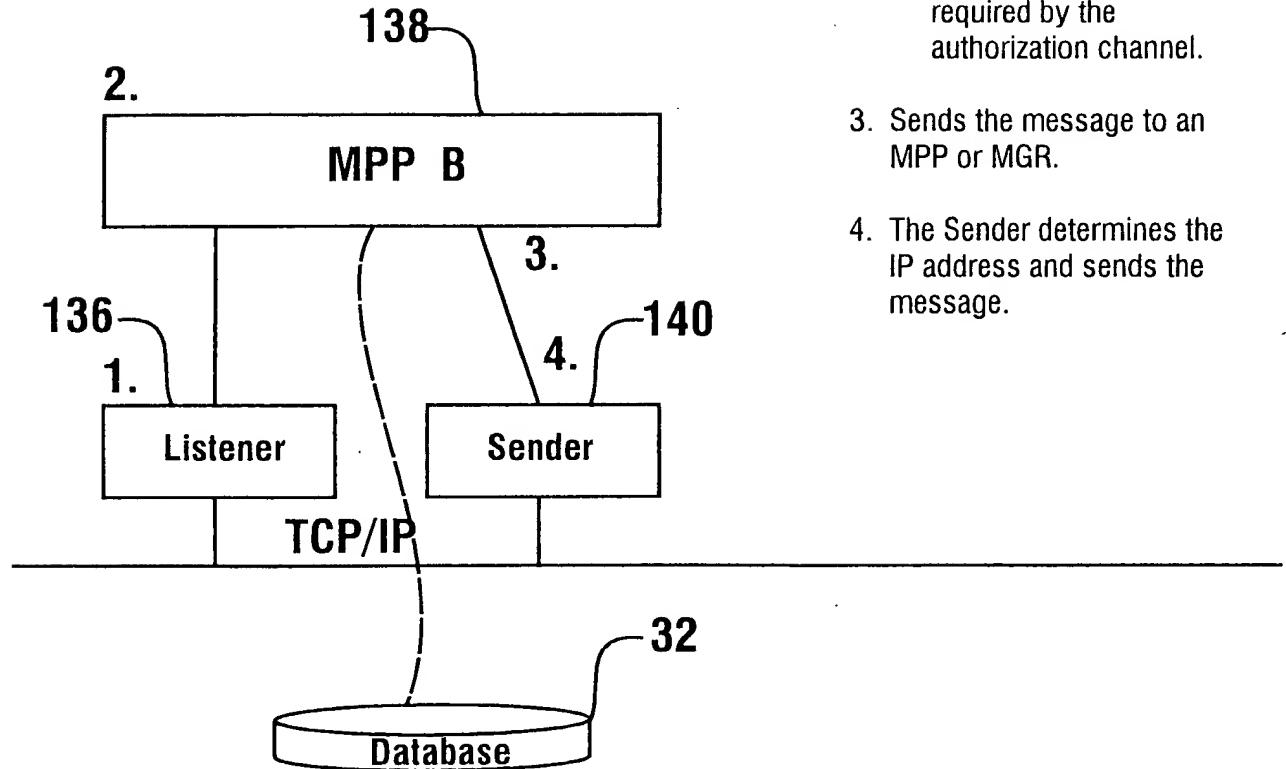


**FIG. 4**



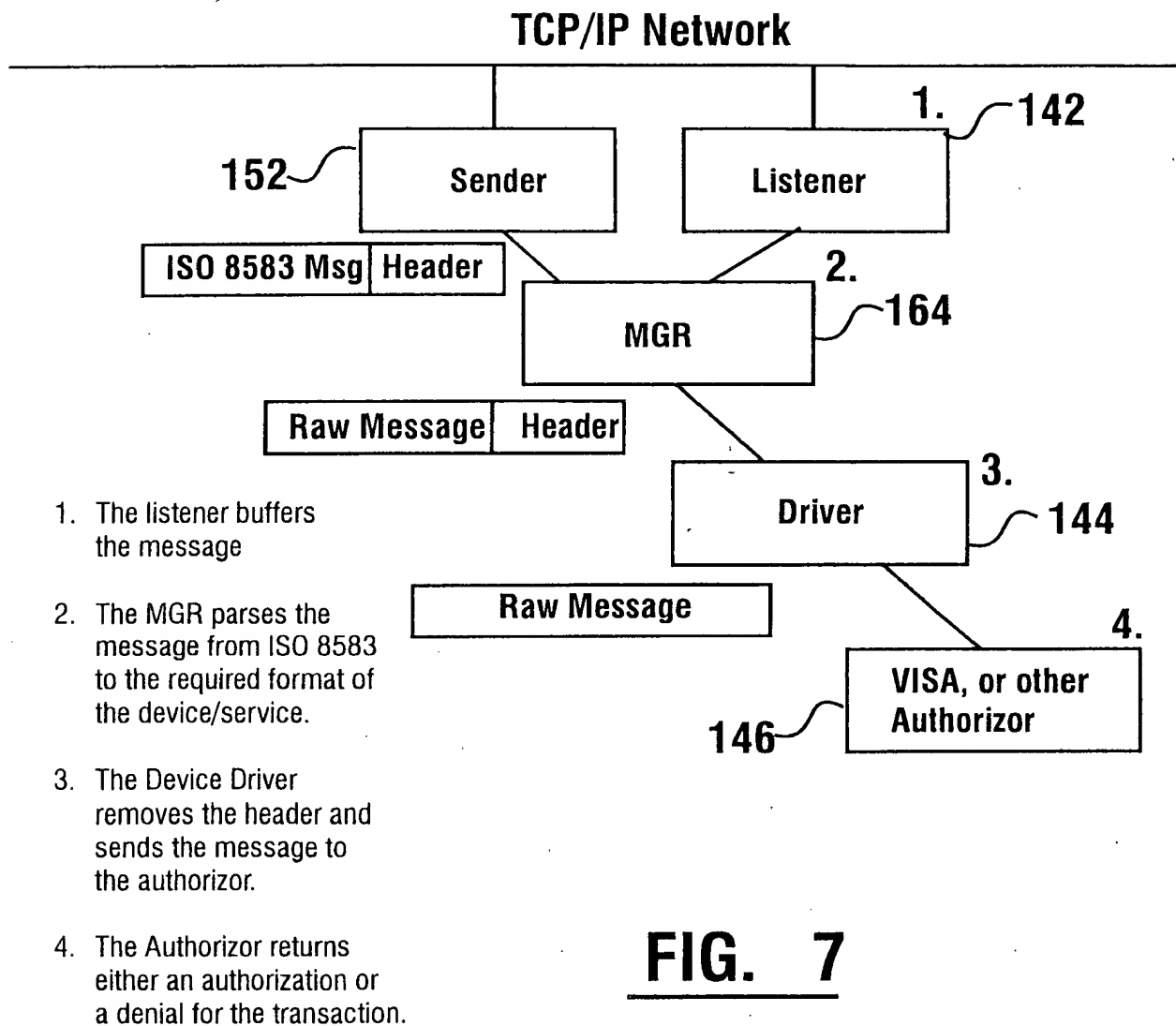
1. The Listener buffers the data, then places the data onto the input queue of the MPP.
2. The MPP performs various functions based upon the requirements of the message.
  - Builds an internal array.
  - Parses composite fields into subfields of the array.
  - May perform authorization.
  - Determines who to send the message to. May be an MPP or MGR
  - Builds a new message.
3. Sends a copy of the data to the database for archive.
4. Sends the message to the authorization host.
5. The Sender determines the IP address and sends the message.

**FIG. 5**



1. The Listener buffers the data, then places the data onto the input queue of the MPP.
2. The MPP performs various functions based upon the requirements of the message.
  - Builds an internal array.
  - Builds any subfields required by the authorization channel.
3. Sends the message to an MPP or MGR.
4. The Sender determines the IP address and sends the message.

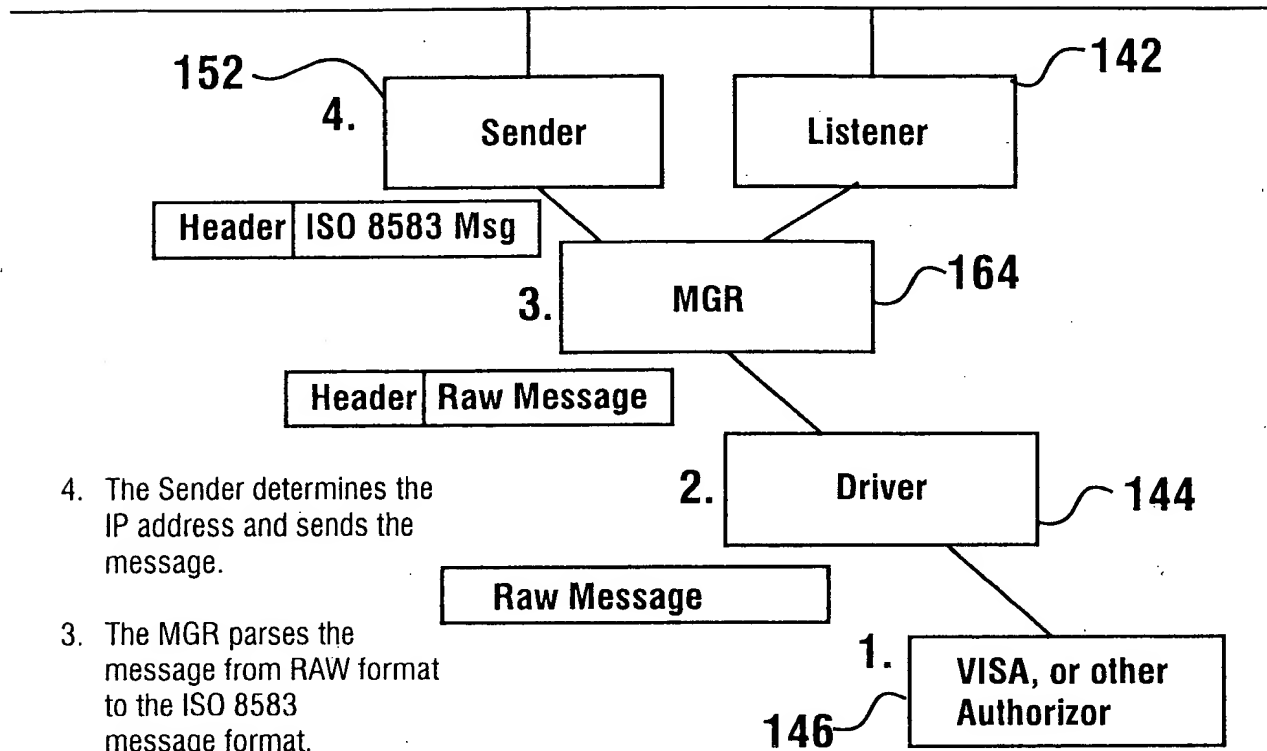
**FIG. 6**



**FIG. 7**

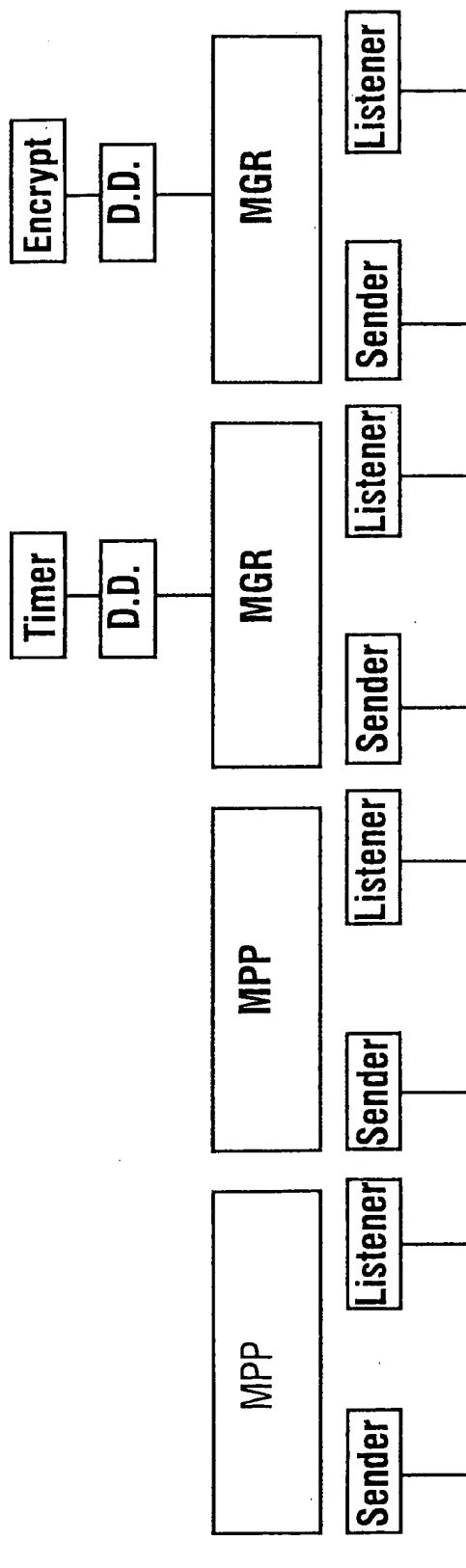


## TCP/IP Network



4. The Sender determines the IP address and sends the message.
3. The MGR parses the message from RAW format to the ISO 8583 message format.
2. The device driver add a header and fills some fields after getting the information directly from the network or host.
1. The Authorizer (VISA or other host) returns the message. This represents the actual host/network.

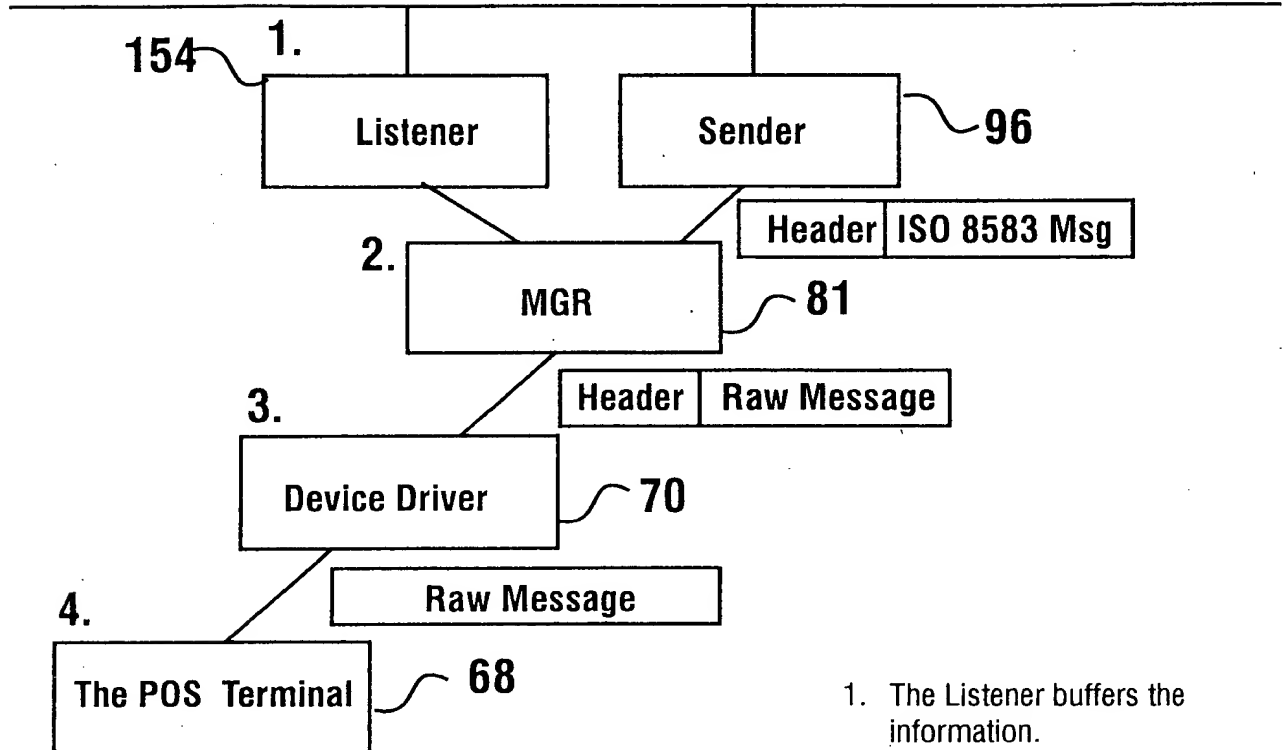
**FIG. 8**



1. The Message is sent to the second MPP. It uses a echo-back field to determine the origin of the message. The database contains the original message with a key. It may send the message to the first MPP by calling the Encryption Device for decryption of the PAN.
2. The message is received by the first MPP. It may need to build special fields, such as track II data. It will then send the message back to the original calling device by using the saved data in the database.

**FIG. 9**

## TCP/IP Network



**FIG. 10**

1. The Listener buffers the information.
2. The MGR parses the ISO 8583 message into a message format that can be used by the device.
3. The Device driver sends the message to the device.
4. The POS terminal returns a message confirming the authorization message.

The message is then returned to the MPP in the same manner as before.

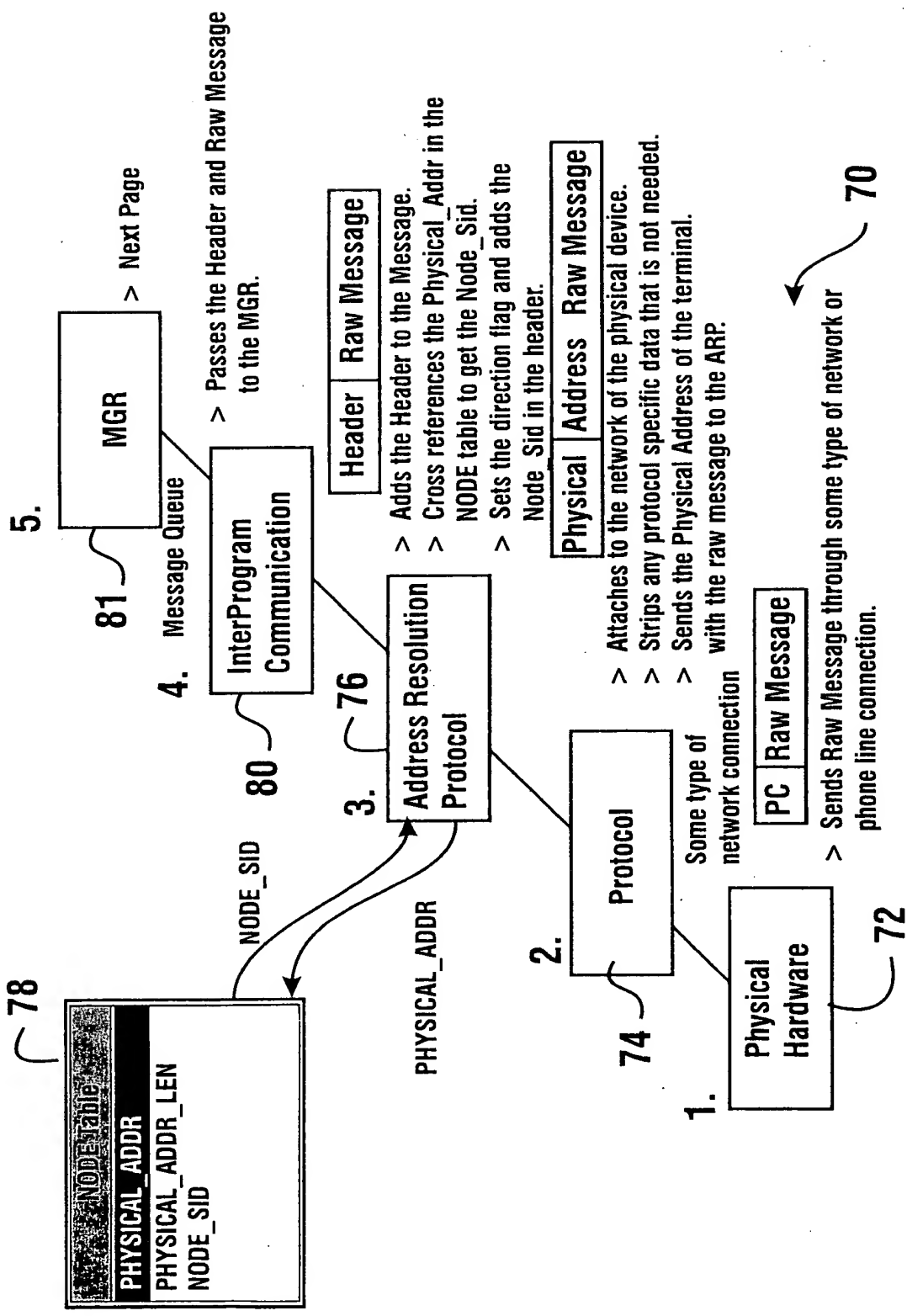
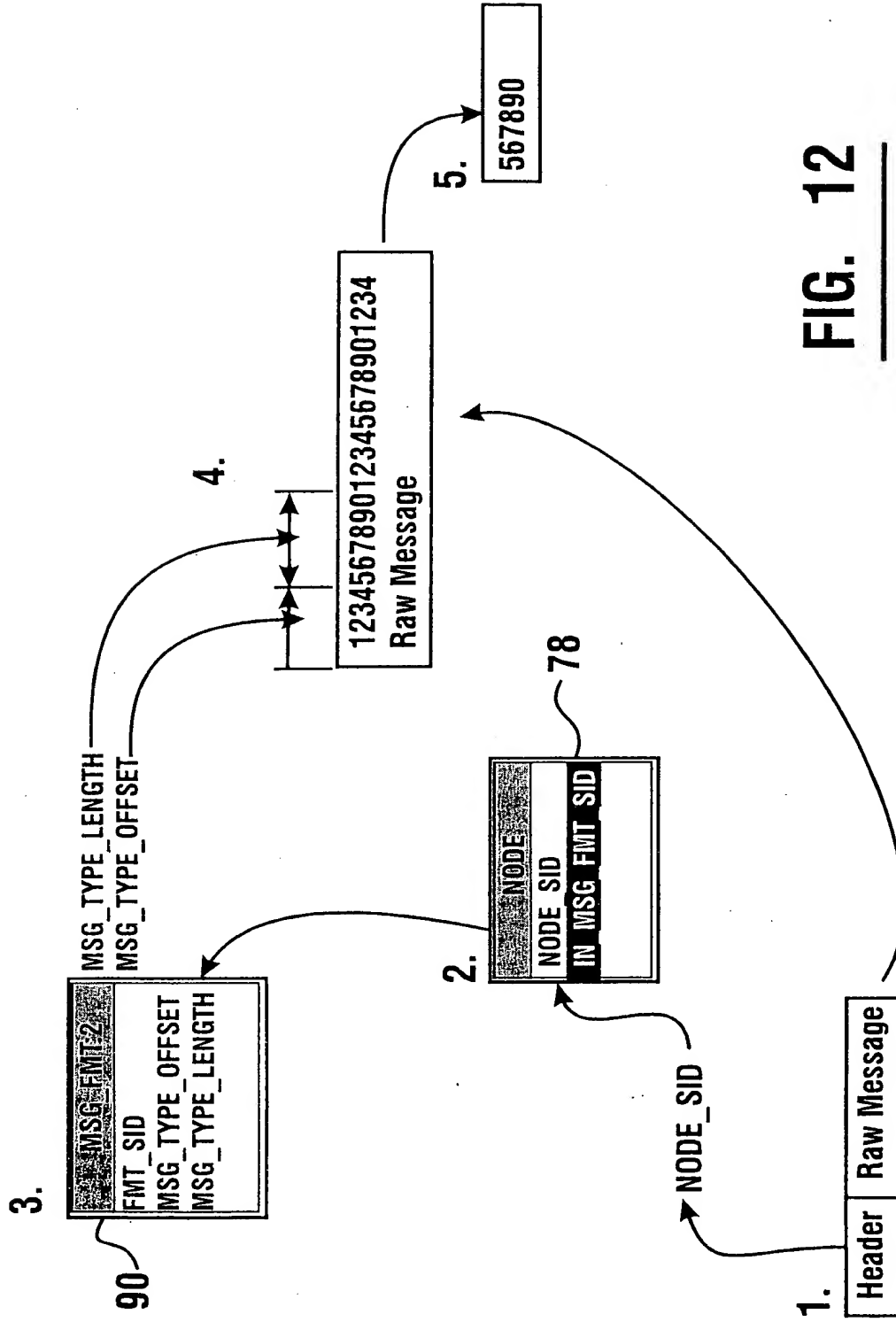
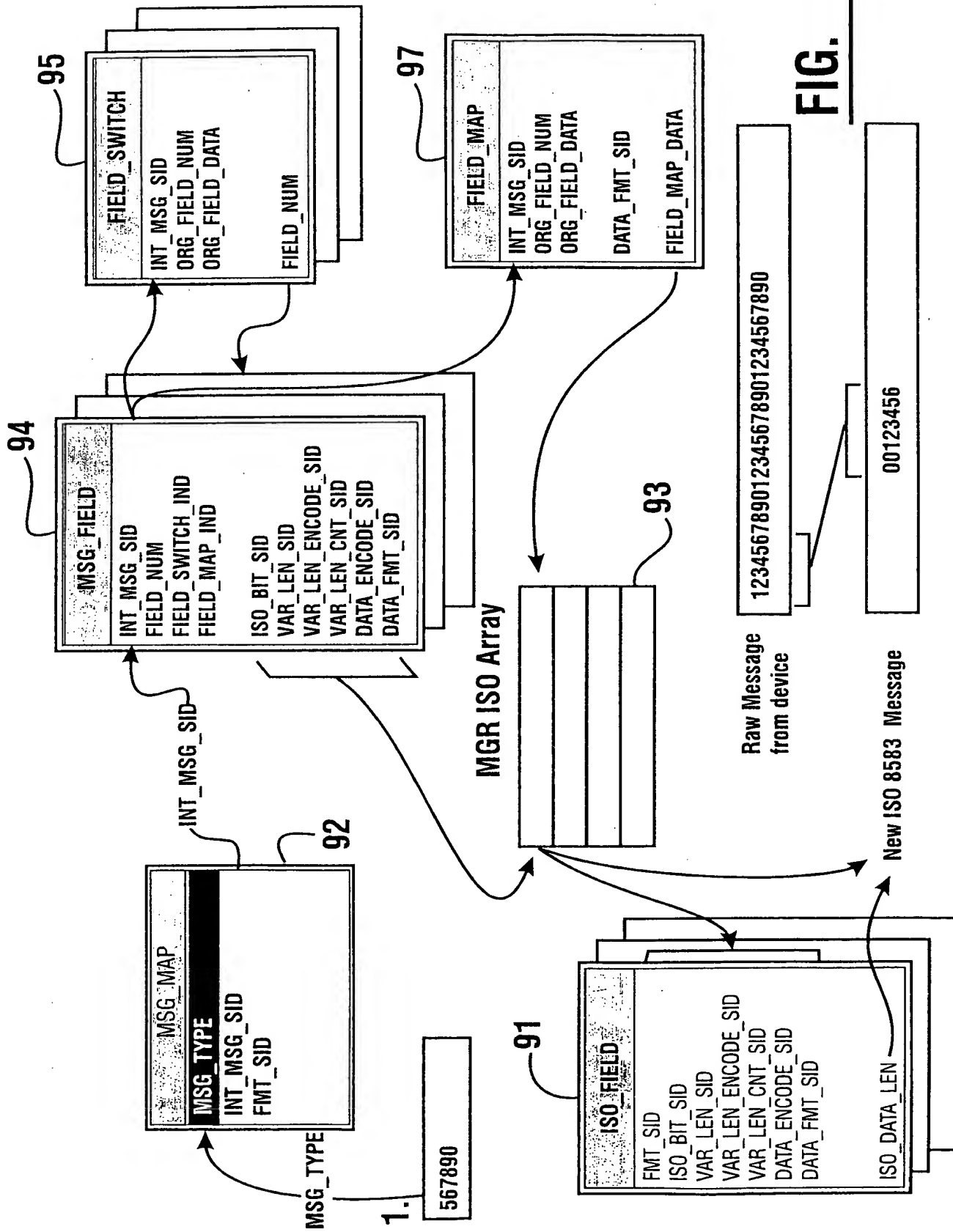
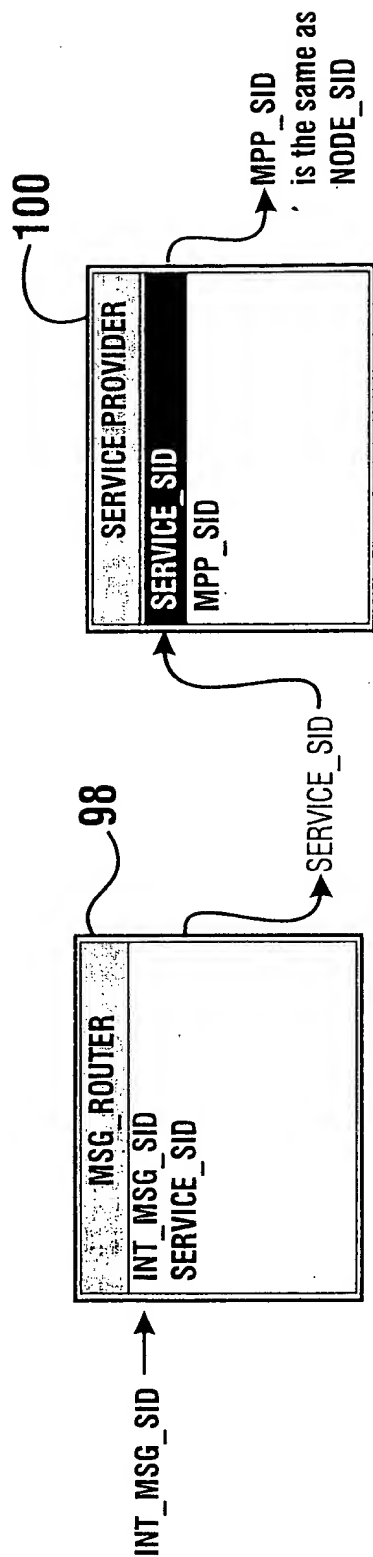


FIG. 11



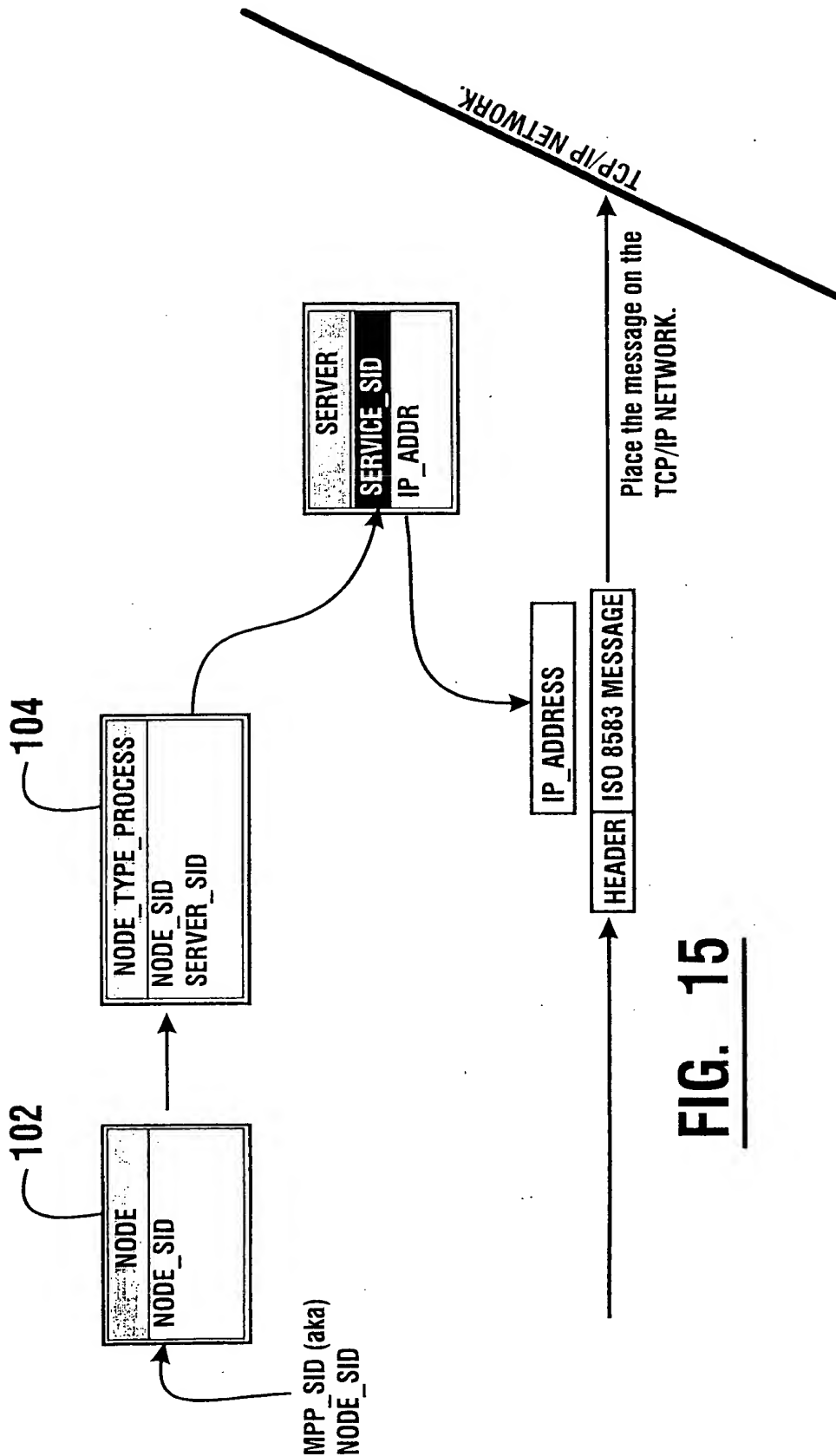
**FIG. 12**





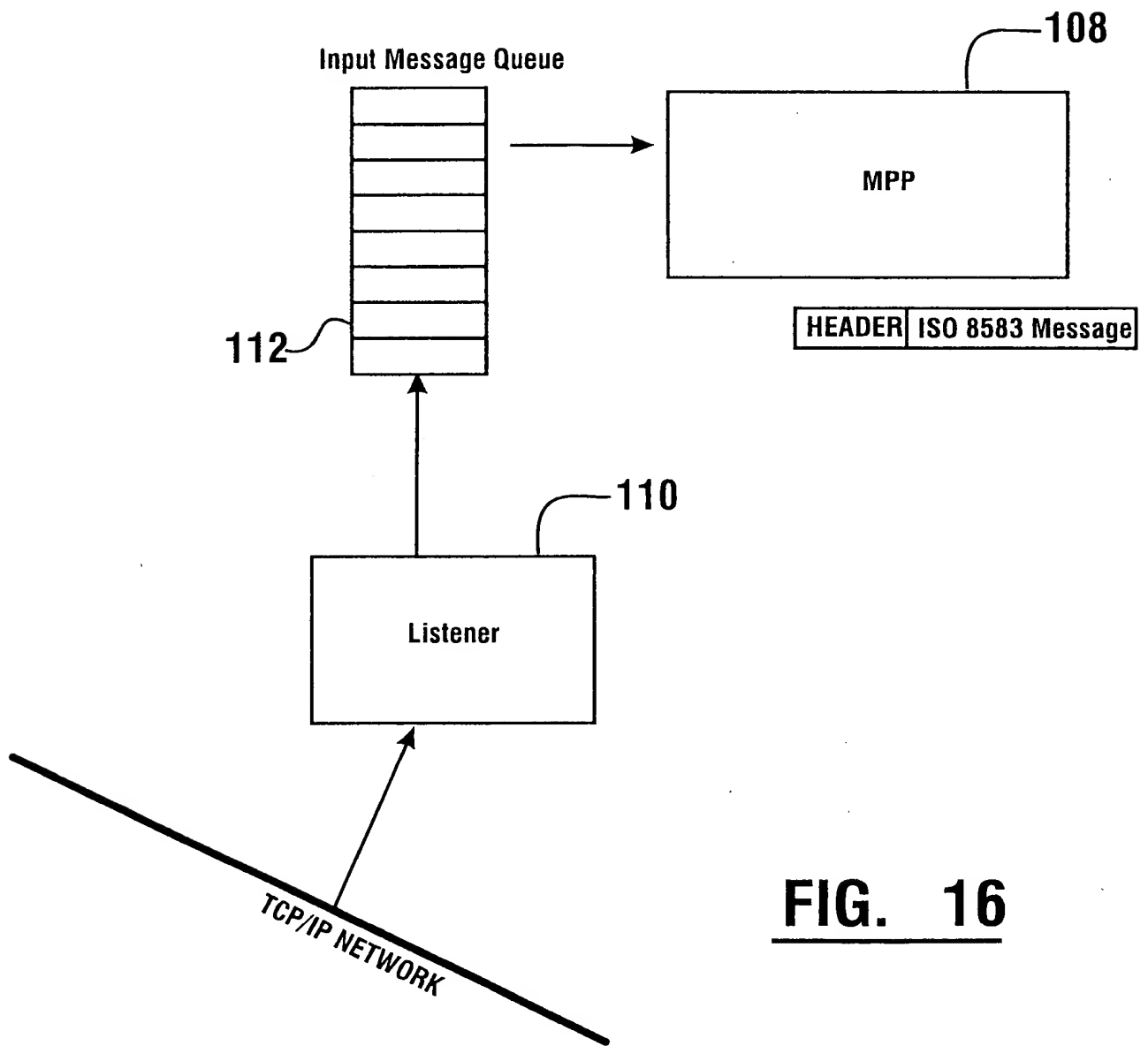
If the chosen Provider is not available at the time of the TCP/IP call. This table is used to determine if there is another service provider. Hot spare - fault tolerance.

**FIG. 14**



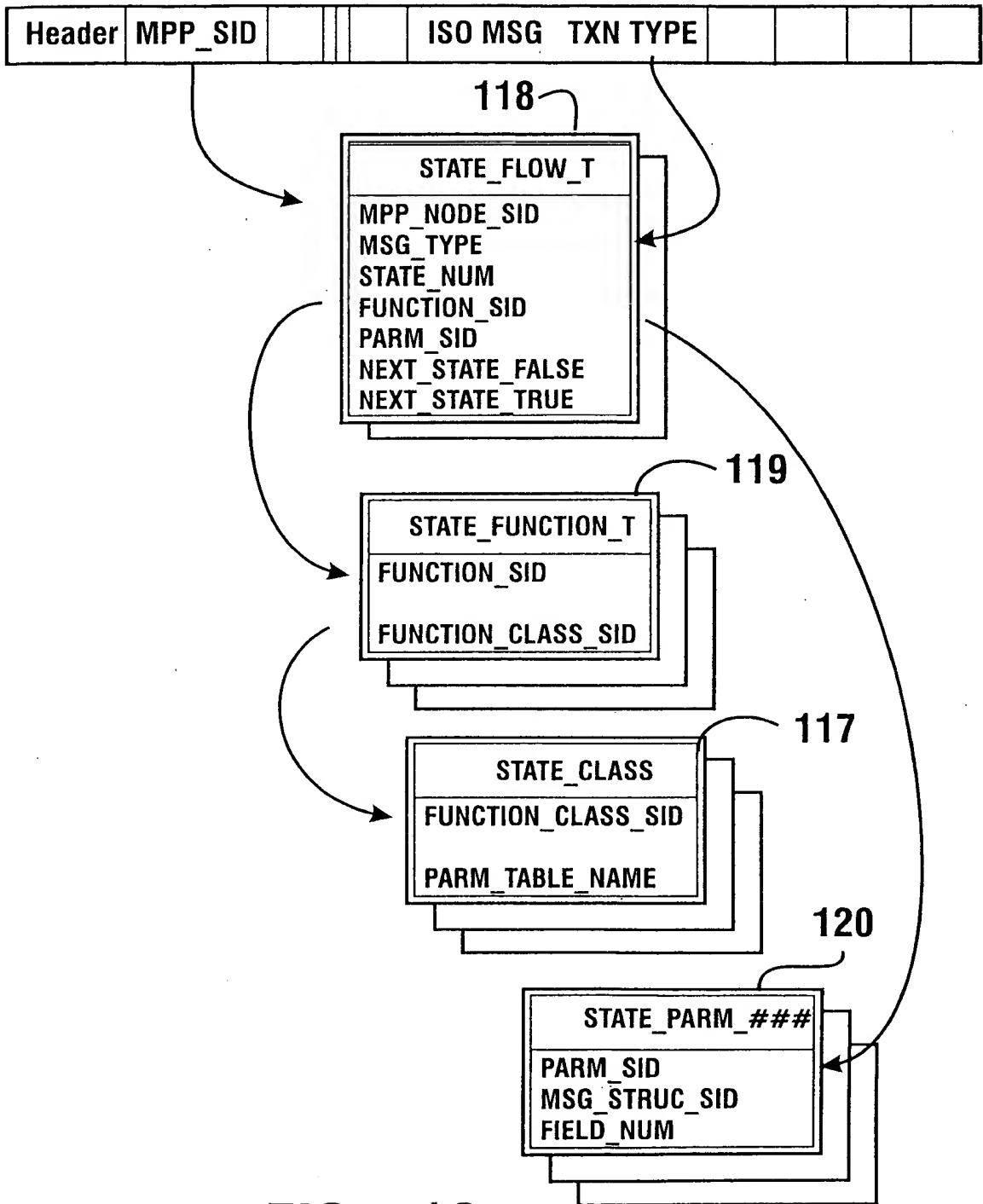
**FIG. 15**





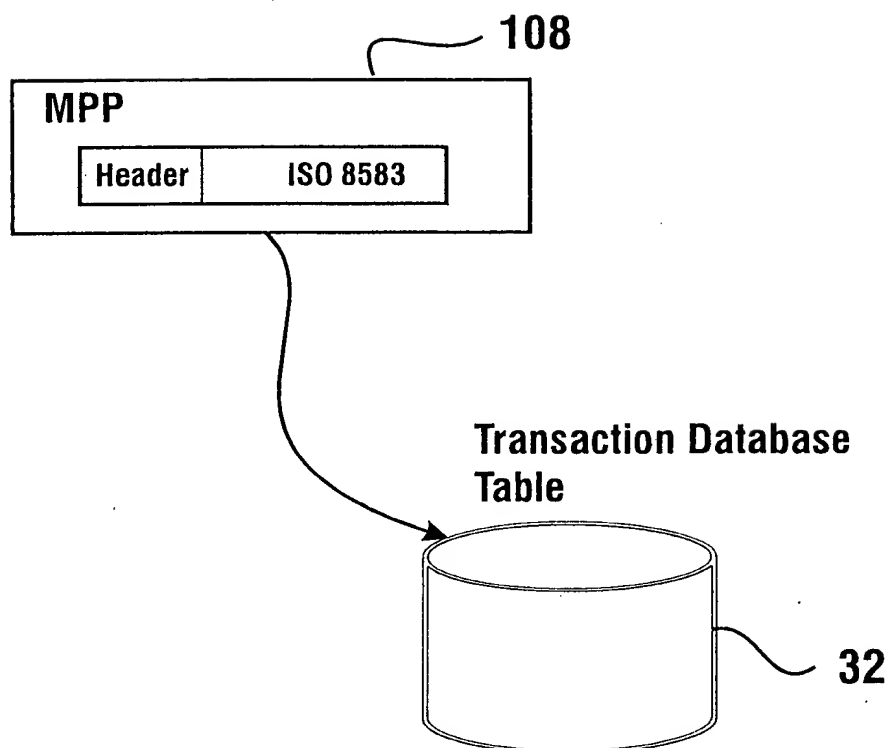
**FIG. 16**





**FIG. 18**

09867183.052904  
T06250 EST 9860



**FIG. 19**

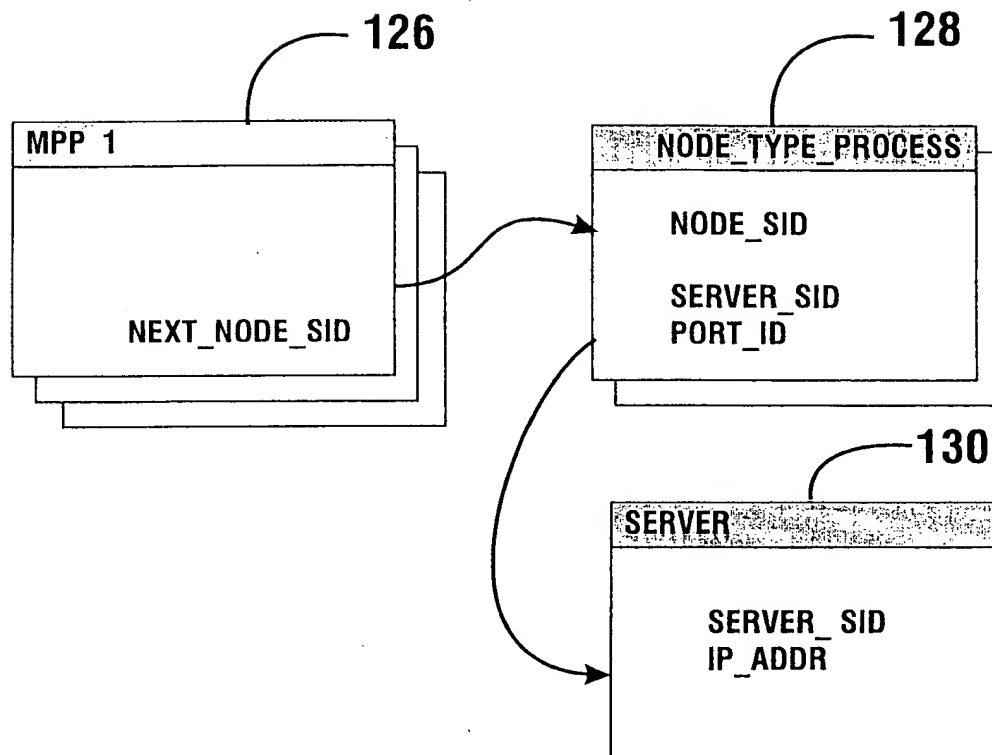


FIG. 20

**OBJECT\_SID = TARGET\_OBJECT\_SID**

SNMP	
OBJECT_SID	not null
OBJECT_TYPE	not null
OBJECT_ID	not null
TARGET_OBJECT_SID	not null
TARGET_OBJECT_ID	not null
STATUS	CHAR(3)
REASON_MASK	CHAR(10)
STATUS_DATE	DATE
FUNCTION	NUMBER(4)
IP_ADDR	VARCHAR2(20)
PORT_ID	NUMBER(6)
BUFFER_DATA	VARCHAR2(255)
BUFFER_LEN	NUMBER(3)
USER_SID	VARCHAR(6)

**NODE\_TYPE\_SID = NODE\_TYPE\_SID**

NODE_TYPE	
NODE_TYPE_SID	<PK> NUMBER(4) not null
DESCRIPTION	VARCHAR2(30) not null
DISPLAY_IND	CHAR(1) null


```
V_MONITOR_MODE
node_sid
node_num
node_type_sid
activity_count
sess_status
sess_status_date
```

[illegible]

TERM\_TYPE\_SID = TERM\_TYPE\_SID

TERM_TYPE			
TERM_TYPE	SID	<pk>	NUMBER(4)
DESCRIPTION			VARCHAR2(30)
DISPLAY_IND			CHAR(1)
			not null
			not null
			not null

[illegible]

VIEW_TERM	NUMBER(4)
node.parent_node_id	
node_node_id	
node_in_msgg_term_id	
ref_term_type_id	
node_and_msgg_term_id	
node_node_type_id	
 node_type_term_int	

V_LINE_GRP_COUNT
Count(distinct(line_grp_node_id)) group_count
parent_node_id node_id

[illegible][illegible]



LINE	TYPE	DESCRIPTION	DISPLAY	IND
<pk>	NUMBER(4)			not null
	VARCHAR2(30)			not null
	CHAR(1)			not null



	MODE_SID	PROCESS_TYPE_SID	SID <IN>	<PL_H>NUMBER(6)	not null
SERVER SID	NUMBER(4)			NUMBER(4)	not null
IP_PORT_ID	NUMBER(6)			NUMBER(6)	not null
MIN_THREAD	NUMBER(2)			NUMBER(2)	not null
MAX_THREAD	NUMBER(2)			NUMBER(2)	not null
PROCESS_PRIORITY	CHAR(3)			CHAR(3)	not null
DEVICE_TYPE	VARCHAR2(30)			VARCHAR2(30)	not null
DRIVER NAME	VARCHAR2(80)			VARCHAR2(80)	not null
SENDER NAME	VARCHAR2(30)			VARCHAR2(30)	not null
LISTENER NAME	VARCHAR2(20)			VARCHAR2(20)	not null
CONTEXT_TABLE_NAME	NUMBER(6)			NUMBER(6)	not null
EPP_PRIORITY	NUMBER(6)			NUMBER(6)	not null
SHMEM_NODE_SID	VARCHAR2(30)			VARCHAR2(30)	not null
CTRL_NAME					

PROCESS_TYPE			
PROCESS_TYPE	SID	<B> NUMBER(4)	not null
DESCRIPTION		VARCHAR2(30)	not null
DISPLAY-IND		CHAR(1)	not null

MODEM	
MODEM SID	<NUMBER(4)
MODEM NUM	<VARCHAR(215)
MODEM NAME	<VARCHAR(215)
CONNECT MSG_LEN	not null
CONNECT MSG	not null
INIT MSG_LEN	not null
INIT MSG	not null
DISCONNECT MSG_LEN	not null
DISCONNECT MSG	not null
HANGUP MSG_LEN	not null
HANGUP MSG	not null
END MSG TOKEN	not null
DESCRIPTION	not null

VIEW_EPP_ADDR	
Ntp_node_sld	NUMBER(6)
server_ip_addr	VARCHAR2(20)
0_app_priority	
<input type="checkbox"/> server	
<input type="checkbox"/> node_type_process ntp	

VIEW_PROCESS	
mp_node_sid	NUMBER(6)
pi_driver_name	VARCHAR2(30)
pi_sendor_name	VARCHAR2(30)
mp_listenername	VARCHAR2(30)
mp_server_sid	NUMBER(4)
mp_ip_port_id	NUMBER(6)
mp_shimem_node_sid	NUMBER(6)
mp_cpu_name	VARCHAR2(30)
node_node_type_sid	
node_parent_node_sid	
node_sess_status	NUMBER(4)
node_process_type_sid	
node_status_data	
	node_type_process mp
	process_type pi

VIEW_SMPs	
node_node_sid	VARCHAR2(20)
server_ip_addr	NUMBER(6)
tcp_ip_port_id_port_id	NUMBER(2)
tcp_process_priority	VARCHAR2(30)
tcp_driver_name	VARCHAR2(30)
tcp_sender_name	VARCHAR2(30)
tcp_listener_name	VARCHAR2(30)
node_in_msg_lmt_sid	
node_out_msg_lmt_sid	
 server	
 node_type_process_rtp	

VIEW_EPP	
rtb_node_id	NUMBER(6)
s_ip_addr	VARCHAR2(20)
rtb_process_priority	NUMBER(2)
rtb_driver_name	VARCHAR2(30)
O_epp_priority	
-rtb_sender_name	VARCHAR2(30)
server \$	
node_type_process rtb	

VIEW_MPP	
<input type="checkbox"/> node.node_id mpp_aid	VARCHAR(20)
<input type="checkbox"/> service_ip_addr	NUMBER(6)
<input type="checkbox"/> ntp_ip_port_id	NUMBER(2)
<input type="checkbox"/> ntp_ip_port_id_port_id	NUMBER(2)
<input type="checkbox"/> ntp_min_thread	NUMBER(2)
<input type="checkbox"/> ntp_max_thread	VARCHAR(20)
<input type="checkbox"/> ntp_process_priority	VARCHAR(20)
<input type="checkbox"/> ntp_context_table_name	VARCHAR(20)
<input type="checkbox"/> ntp_driver_name	VARCHAR(20)
<input type="checkbox"/> ntp_sender_name	VARCHAR(20)
<input type="checkbox"/> ntp_listener_name	VARCHAR(20)
<input type="checkbox"/> server	
<input type="checkbox"/> node_type_process ntp	

**FIG. 21**

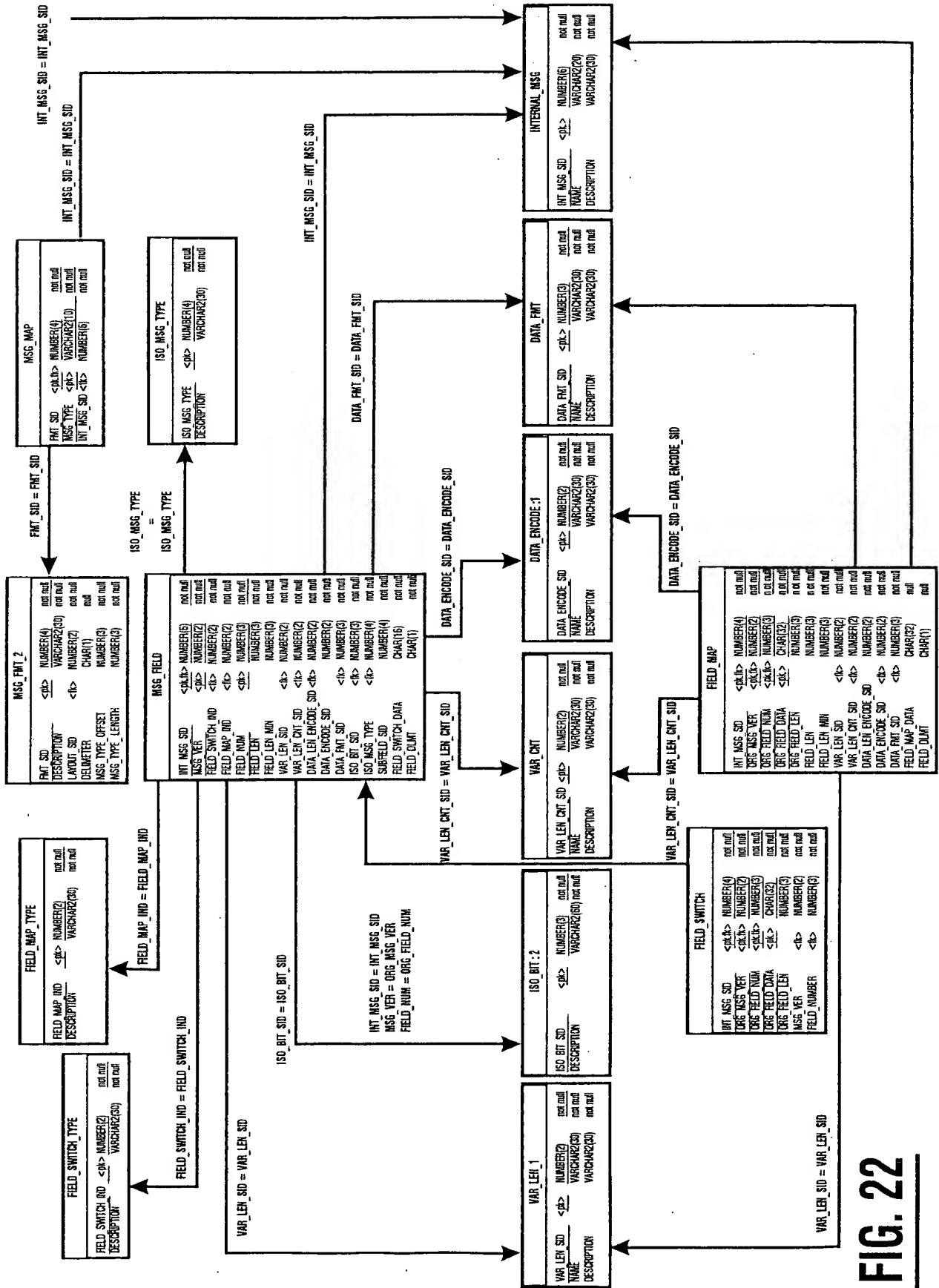


FIG. 22

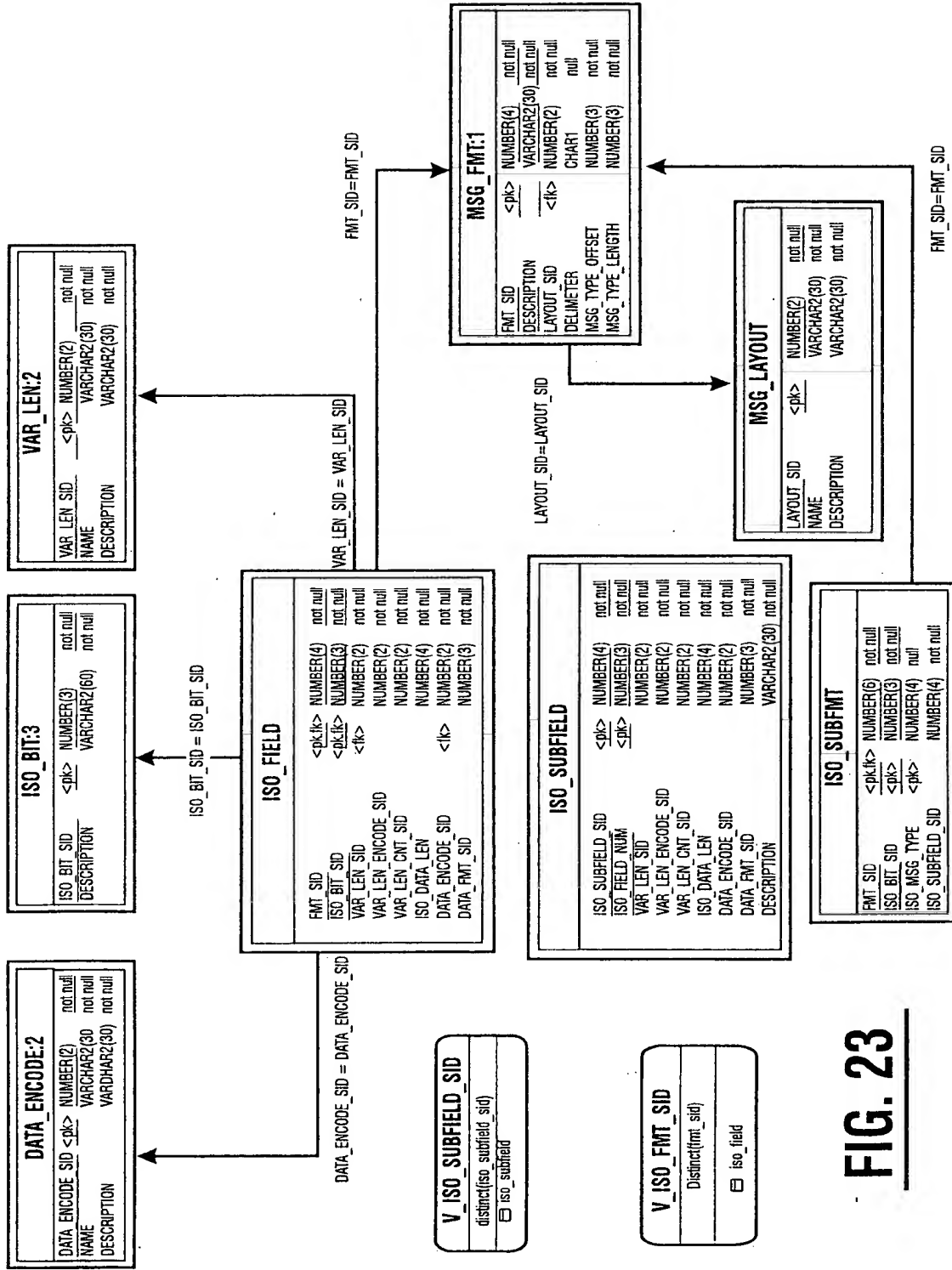


FIG. 23



EXTERNAL_HOST			
HOST_SID	<pk>	NUMBER(6)	not null
HOST_NUM		VARCHAR2(15)	null
NAME		VARCHAR2(30)	null
ADDR		VARCHAR2(30)	null
CITY		VARCHAR2(20)	null
STATE		CHAR(2)	null
COUNTRY_CODE		CHAR(3)	null
ZIP_CODE		CHAR(9)	null
CONTACT_NAME		VARCHAR2(30)	null
TELEPHONE		VARCHAR2(16)	null
NODE_SID		NUMBER(6)	null
COMMENTS		VARCHAR2(30)	null
STATUS		CHAR(3)	null
STATUS_DATE		DATE	null

SERVER_HOST_LINK			
SERVER_SID	<pk.fk>	NUMBER(6)	not null
HOST_SID	<pk.fk>	NUMBER(6)	not null
PRIORITY		NUMBER(2)	null

SERVER\_SID = SERVER\_SID

SERVER			
SERVER_SID	<pk>	NUMBER(6)	not null
NAME		VARCHAR2(20)	not null
IP_ADDR		VARCHAR2(20)	not null

COL_VALUE			
TABLE_NAME	<pk>	VARCHAR2(20)	not null
COLUMN_NAME	<pk>	VARCHAR2(20)	not null
ITEM_OFFSET	<pk>	NUMBER(2)	not null
COLUMN_VALUE	<pk>	VARCHAR2(3)	not null
DESCRIPTION		VARCHAR2(30)	not null

STATUS_REASON			
TABLE_NAME	<pk>	VARCHAR2(20)	not null
STATUS_VALUE	<pk>	CHAR(3)	not null
REASON_NUM	<pk>	NUMBER(2)	not null

SYSTEM_PARM			
SYSTEM_PARM_SID	<pk>	NUMBER(4)	not null
PARAMETER	<pk>	VARCHAR2(10)	not null
VALUE		VARCHAR2(20)	not null
FMT		VARCHAR2(10)	not null
STATUS		CHAR(3)	not null
STATUS_DATE		DATE	not null
DESCRIPTION		VARCHAR2(30)	not null

**FIG. 24**

098743210

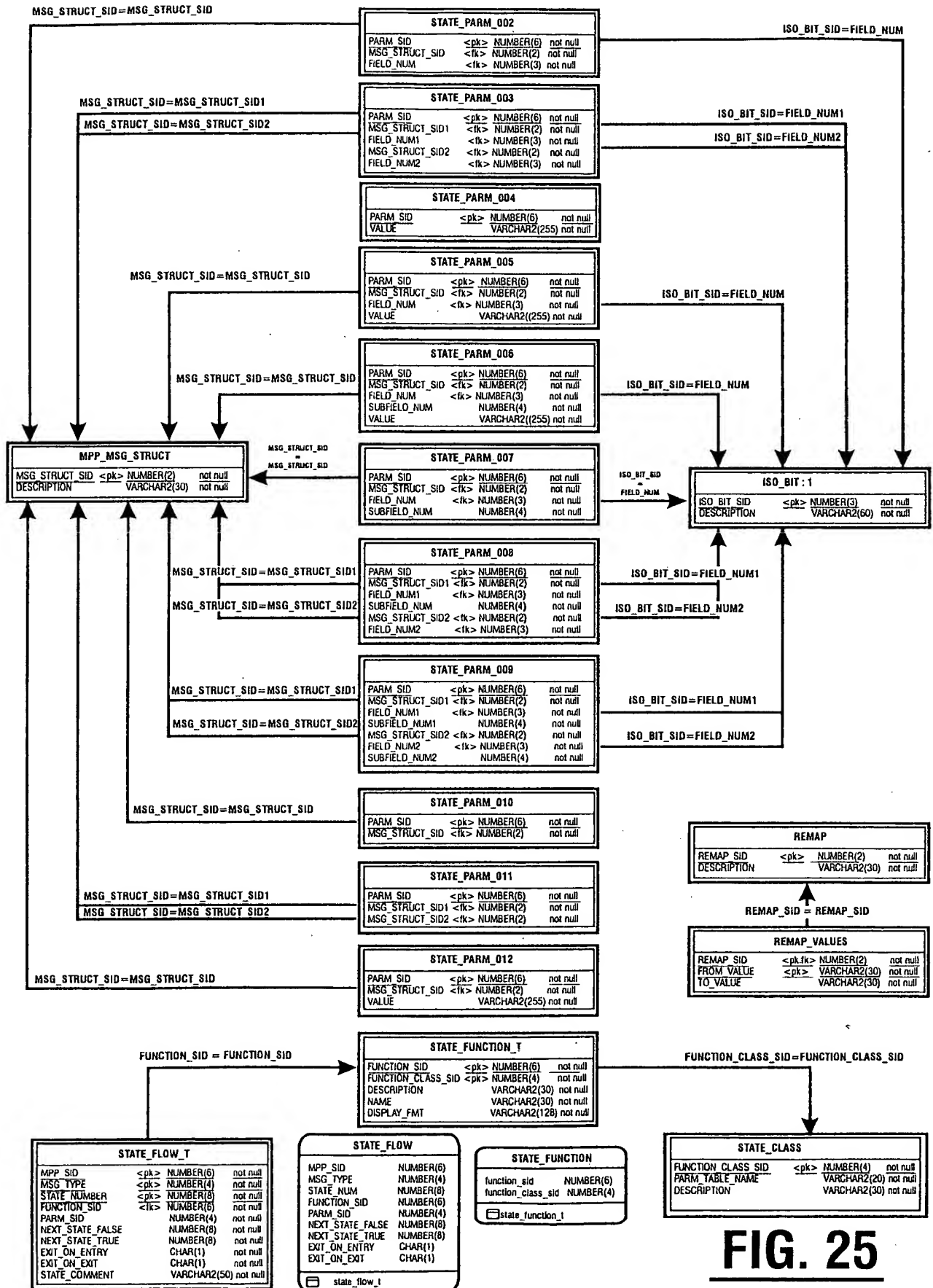
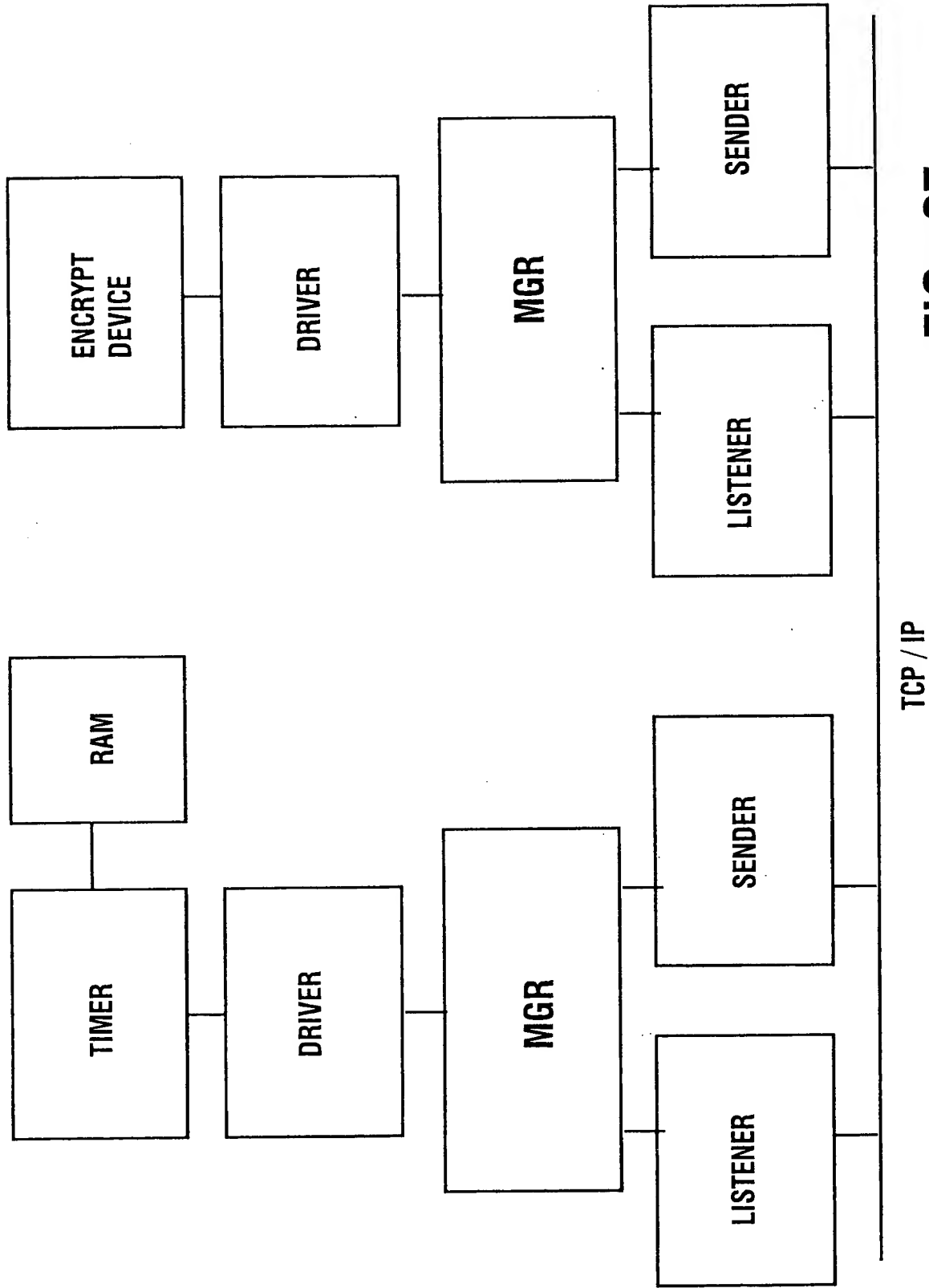


FIG. 25





**FIG. 27**

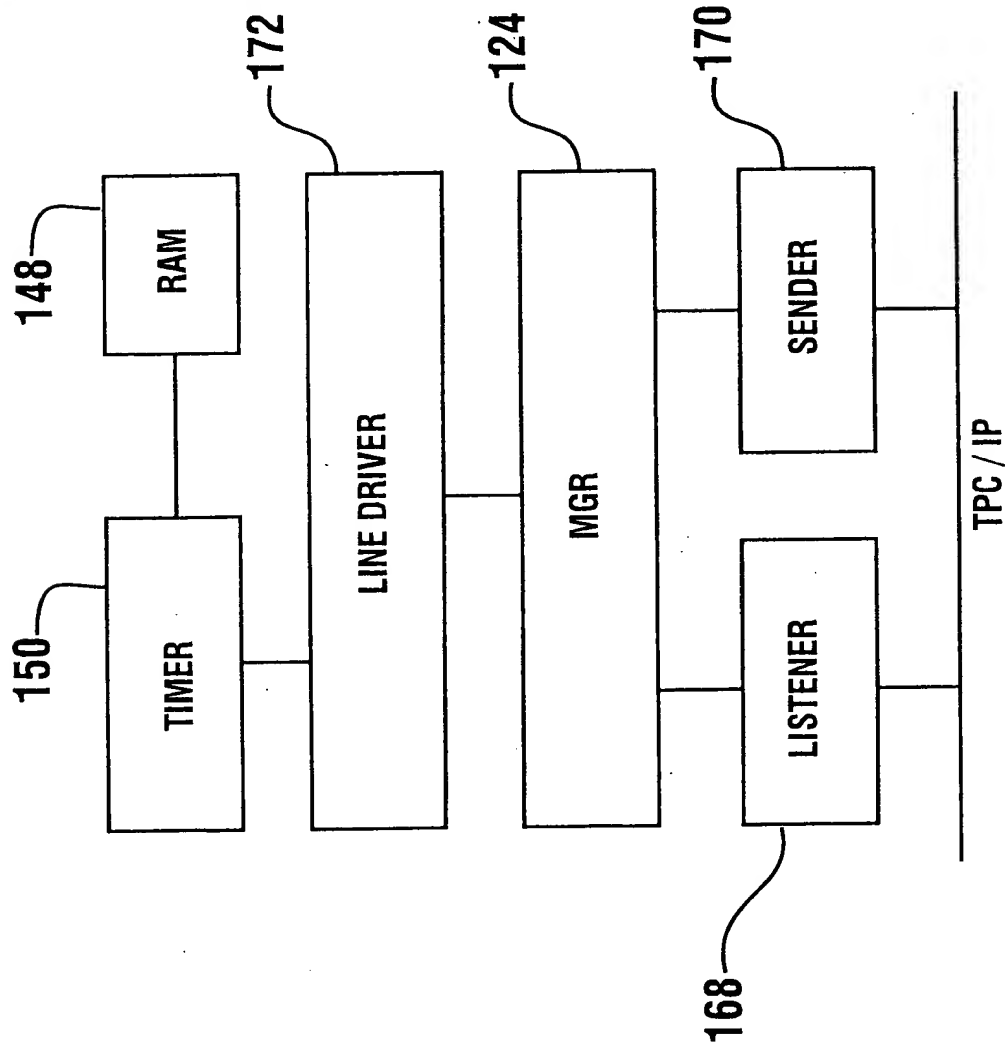


FIG. 28

MGS_ROUTER					
LINE_SID	<pk>	NUMBER(6)	not null		
NODE_SID	<pk>	NUMBER(6)	not null		
INT_MSG_SID	<pk, fk>	NUMBER(6)	not null		
SERVICE_SID	<pk>	NUMBER(4)	not null		

SERVICE\_SID = SERVICE\_SID

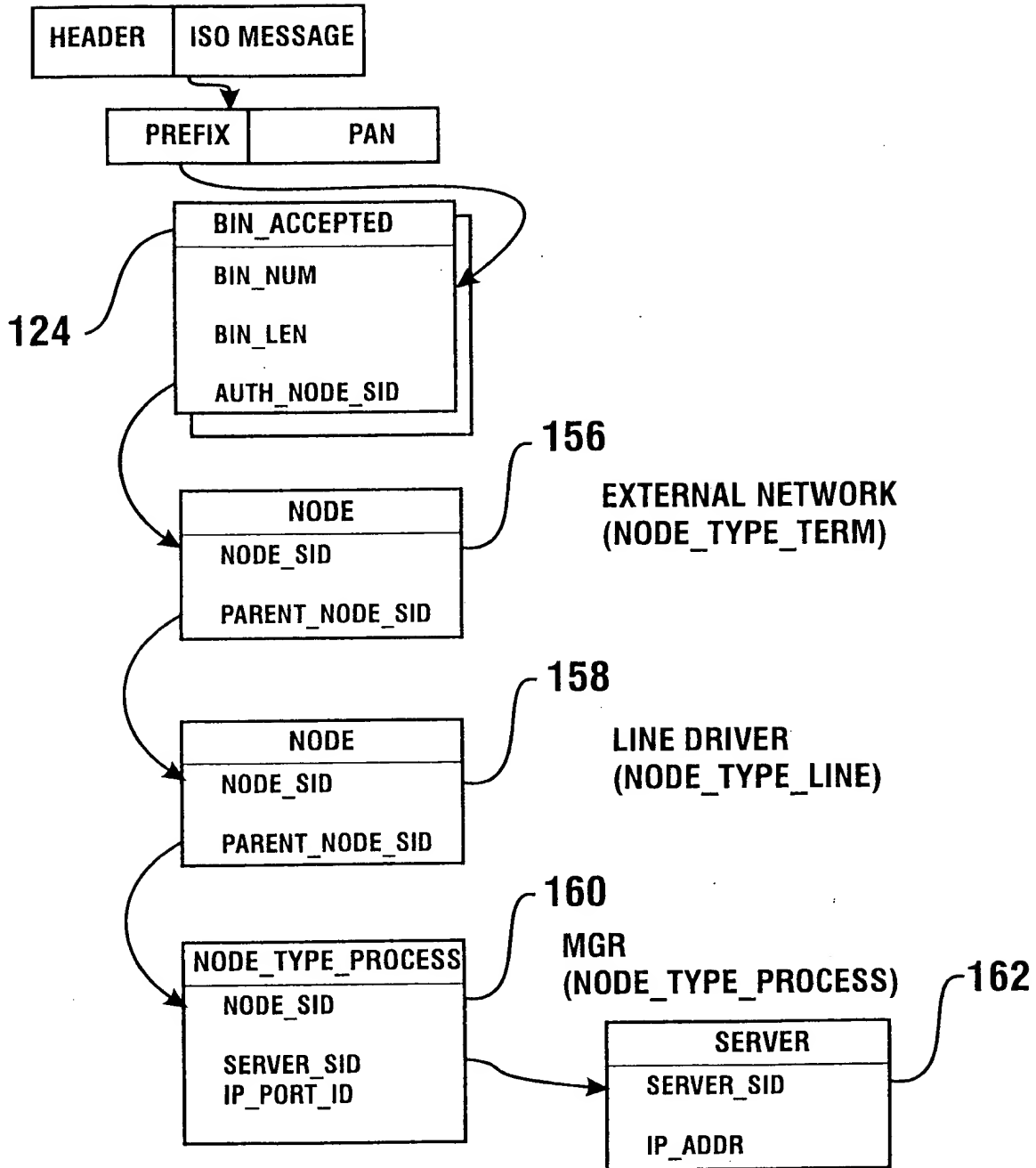
SERVICE			
SERVICE_SID	<pk>	NUMBER(4)	not null
DESCRIPTION		VARCHAR2(30)	not null

SERVICE\_SID = SERVICE\_SID

SERVICE_PROVIDER					
SERVICE_SID	<pk, fk>	NUMBER(4)	not null		
PATH_ORIGINAL	<pk>	NUMBER(2)	not null		
MPP_SID		NUMBER(6)	not null		
PRIORITY		NUMBER(2)	not null		

FIG. 29

0986187 039901  
T06250" EBT 9860



**FIG. 30**